



Programação Procedimental

GBC014 – 2015/2

Prof. Renan Cattelan – www.facom.ufu.br/~renan

Prática 3

Estruturas de Repetição em C

for

```
for (<inicialização>; <teste>; <incremento>)  
    <comando>
```

```
for (<inicialização>; <teste>; <incremento>)  
{  
    <comando 1>  
    ...  
    <comando n>  
}
```

for

Condição testada
antes de cada
iteração

```
for (<inicialização>; <teste>; <incremento>)  
    <comando>
```

Expressão
avaliada antes da
1ª iteração

Expressão
avaliada após
cada iteração

```
for (<inicialização>; <teste>; <incremento>)  
{  
    <comando 1>  
    ...  
    <comando n>  
}
```

for

```
main.c X
1  /* programa ascii.c
2     imprime a tabela ASCII */
3
4  #include "stdio.h"
5
6  int main(void)
7  {
8     int i;
9     for (i=0; i<256; i++)
10        printf("%3d \t %c \n", i, i);
11    return 0;
12 }
13
```

for

```
main.c X
1  /* programa ascii.c
2     imprime a tabela ASCII */
3
4  #include "stdio.h"
5
6  int main(void)
7  {
8     int i=0;
9     for (; i<256; i++)
10        printf("%3d \t %c \n", i, i);
11    return 0;
12 }
13
```

for - aninhamento

```
main.c X
1      /* taboada */
2
3      #include "stdio.h"
4
5      int main(void)
6      {
7          int i,j;
8          for (i=2; i<=9; i++)
9          {
10             for (j=1; j<=9; j++)
11                 printf("%d x %d = %2d \n", i, j, i*j);
12             printf("\n");
13         }
14     }
15
```

while

```
<inicialização>  
while (<teste>)  
{  
    <comandos>  
    <incremento>  
}
```

while

```
main.c X
1  #include <stdio.h>
2  #include <stdlib.h>
3  int main()
4  {
5      int i=0;
6      while (i < 256)
7      {
8          printf("%3d \t %c \n", i, i);
9          i++;
10     }
11     return 0;
12 }
13
```

%d

0

1

2

...

9

10

11

...

100

%3d

0

1

2

...

9

10

11

...

100

do-while

```
do
{
    <comandos>
    <incremento>
} while (<teste>);
```

do-while

```
main.c X
1  #include <stdio.h>
2
3  int main()
4  {
5      int i=0;
6      do
7      {
8          printf("%3d \t %c \n", i, i);
9          i++;
10     } while (i < 256);
11     return 0;
12 }
13
```

Resumo

- ❑ Comandos de repetição: for, while e do-while
- ❑ Usados quando blocos de comandos precisam ser executados repetidamente
- ❑ for e while, testam a condição de parada e então executam
- ❑ do-while primeiro executa e só então verifica a condição de parada

Exercício

- ❑ Elabore um programa que solicite um número inteiro ao usuário e crie um novo número inteiro com os dígitos em ordem inversa. Por exemplo, uma execução do programa é:
 - ❑ Digite um número inteiro: 5382
 - ❑ Seu número invertido é: 2835

```
1  #include<stdio.h>
2
3  int main(void)
4  {
5      int numero, resto;
6      printf("Digite um numero inteiro: ");
7      scanf("%d", &numero);
8      printf("\nSeu numero invertido e: ");
9      if (numero == 0)
10         printf("%d", numero);
11     while(numero > 0)
12     {
13         resto = numero % 10;
14         printf("%d", resto);
15         numero = numero / 10;
16     }
17     printf("\n\n");
18     system("pause");
19     return 0;
20 }
21
```

Exercício

- ❑ Escreva um programa que calcule a soma dos n primeiros termos da sequência de Fibonacci. O valor de n deve ser fornecido pelo usuário.

❑ 0 1 1 2 3 5 8 13 21

```
1  #include<stdio.h>
2
3  main()
4  {
5      int n, prim = 0, seg = 1, prox, c, soma = 0;
6
7      printf("Digite o nro de termos: ");
8      scanf("%d",&n);
9
10     printf("\n\nOs primeiros %d termos da sequencia de Fibonacci sao:\n", n);
11
12     for (c=0; c<n; c++) {
13         if (c <= 1)
14             prox = c;
15         else {
16             prox = prim + seg;
17             prim = seg;
18             seg = prox;
19         }
20         printf("%d ", prox);
21         soma = soma + prox;
22     }
23     printf("\nO valor da soma da sequencia eh: %d\n\n", soma);
24     return 0;
25 }
26
```